REMARKS

Applicant respectfully requests reconsideration of the present application in view of the remarks below.

Claims 1 - 22 are pending in the application. Claims 1 – 12 are withdrawn. Claims 13-22 are rejected. Of these claims, claims 13 and 18 are independent claims, claims 14 – 17 depend either directly or indirectly from claim 13, and claims 19 – 22 depend either directly or indirectly from claim 18.

I. REJECTIONS UNDER 35 U.S.C. § 102 (b)

The Examiner has rejected Claims 13-22 under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 6,684,261 to Orton et al. (hereinafter "Orton"). Applicants traverse this rejection at least because Orton does not disclose "a method of scheduling processing in a hardware threaded circuit, comprising . . . receiving inputs corresponding to unthreaded processing of an application; receiving information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined; and generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing," as set forth in claim 13.

The present invention is directed to a hardware threading mechanism providing temporary borrowing of unutilized pipeline stages from processing elements to boost throughput performance and/or to reduce the power consumption of tasks running on active processing elements (see application as-filed, specification at page 5, line 31 - page 6, line 3). For example, the hardware threading may include dynamic borrowing of available processing resources between interconnected processing elements (see application as-filed, specification at page 2, lines 5 - 13, and claim 13).

Applicants submit that Orton does not describe or contemplate "a method of scheduling processing in a hardware threaded circuit," as recited in claim 13. Instead, Orton describes an object-oriented wrapper for a procedural operating system. See Orton at col. 5, lines 35 - 53. For example, a program can execute using the disclosed object-oriented techniques by accessing code in the object-oriented wrapper, which handles the conversion of object-oriented calls to procedural calls.

The Examiner directs Applicants' attention to Orton at FIG. 4, col. 7, line 50 to col. 8, line 31 for describing "receiving inputs corresponding to unthreaded processing of an application." While these passages may describe "thread classes 404 for enabling an application to access in an object-oriented manner operating system services to spawn, control, and obtain information relating to threads," such a generic description of thread classes for object-oriented thread control in a native procedural operating system environment in no way discloses the claimed "receiving inputs corresponding to unthreaded processing of an application."

With further regard to claim 13, the Examiner directs Applicants' attention to FIGS. 4 and 11, and col. 32, line 33 to col. 33, line 24 and background, for describing "receiving information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined," as set forth in claim 13. While these passages may describe a "code library 100 implements an object-objected class library 402" (see Orton at col. 7, lines 16 - 17), and various thread and scheduling classes, nowhere is there mention of "receiving information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined," as in claim 13.

In particular, Orton discloses generic scheduling classes such as a TThreadSchedule concrete base class that embodies a scheduling behavior of a class, a TIdleThreadSchedule concrete subclass for threads running during system idle, a TServerSchedule concrete subclass for server threads, a TUserInterfaceSchedule concrete subclass for user interface handling, a

TApplicationSchedule class to support long-running threads, and a TPseduoRealTimeThreadSchedule to indicate thread urgency. See Orton at col. 32, line 36 through col. 33, line 16. Orton, therefore, describes various scheduling base classes and subclasses that may be of use in an object-oriented system wrapper for a procedural operating system.

More to this point, it is clear than none of the above-mentioned classes contemplate "receiving information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined." Advantageously, the claimed invention may traverse downstream states in a schedule to look ahead by one or more states based on the window size (see application as-filed, e.g. the specification at page 14, line 31 through page 15, line 3). Future operations can be rescheduled earlier (such as within a current state) by maximizing the usage of underutilized resources.

Moreover, Orton does not describe "generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing", as set forth in claim 13. The claimed invention may use such information to schedule more operations within a current state of an unthreaded schedule (see application as-filed, specification at page 15, lines 5 - 7). For example, if a next state can be scheduled within a current state, such as when next state can be scheduled by borrowing an unused resource, the two states are scheduled together and the states are said to be threaded together (see application as-filed, specification at page 15, lines 8 - 10).

The Examiner directs Applicants' attention to Orton at FIG. 4 col. 11, line 32 to col. 12, line 38, and col. 21, line to col. 23, line 11 for describing "generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing," as set forth in claim 13.

While these passages may describe scheduling classes and thread classes, nowhere is there mention of the claimed features. For example, Orton at col. 21, lines 25 - 30 states:

> "TThreadHandle is a concrete class that represents a thread entity in the system. It provides the methods for controlling and determination about the thread. It also provides the mechanism for spawning new threads in the system. Control operations include killing, suspending/resuming, and doing a death watch on it."

Therefore, while Orton may describe basic thread functionality, such as spawning, killing, and suspending/resuming threads, it does not describe or contemplate the claimed "generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being interconnected to enable dynamic resource sharing." For example, nowhere does Orton describe the claimed "at least first and second one of the processing elements" or "being interconnected to enable dynamic resource sharing."

Furthermore, Applicants submit that Orton is concerned with *distinctly different* problems than with the problems addressed by the claimed invention. In particular, Orton is concerned with an object-oriented system wrapper for allowing object-oriented calls within a native procedural operating system environment (see Detailed Description of Orton at col. 5, lines 35 - 45). Clearly, these problems are different than the problems associated with scheduling processing in a hardware threaded circuit. Therefore, Applicants submit that one of ordinary skill in the art faced with the problems addressed by the claimed invention would not look toward Orton for a solution.

Accordingly, Applicants submit that Orton does not describe "a method of scheduling processing in a hardware threaded circuit, comprising . . . receiving inputs corresponding to unthreaded processing of an application; receiving information including processing element resources, a number of processing elements, and a window size corresponding to a number of downstream processing states to be examined; and generating a hardware threaded schedule for processing the application with at least first and second one of the processing elements being

interconnected to enable dynamic resource sharing," as set forth in claim 13. Therefore, Applicants submit that claim 13 is patentable over Orton and request withdrawal of the art rejections. For at least the same reasons, Applicants submit that claims 14 – 17 are also patentable over Orton and request withdrawal of the art rejections.

Applicant submits that dependent claims contain additional subject matter that further distinguishes over the art of reference.

With regard to claim 14, Orton fails to anticipate "synthesizing the hardware threaded schedule to an Application Specific Circuit (ASC)."

With regard to claim 15, Orton fails to anticipate "synthesizing the hardware schedule to maximize throughput."

With regard to claim 16, Orton fails to anticipate "synthesizing the hardware threaded schedule to reduce power consumption."

With regard to claim 17, Orton fails to anticipate "receiving resource constraint information for the processing elements."

With regard to independent claim 18, Applicants submit that Orton does not describe or contemplate "a hardware threaded circuit system, comprising: . . . a plurality of processing elements coupled to the task manager, wherein first and second ones of the plurality of processing elements are interconnected for hardware threaded processing to enable dynamic borrowing of processing resources associated with the second one of the plurality of processing elements by the first one of the plurality of processing elements," as recited in claim 18. The Examiner alleges that Orton teaches these features at FIG. 4, col. 11, line 32 to col. 12, line 38, and col. 21, line 9, to col. 23, line 11. As described above, these passages describe threads and

thread classes for providing an object-oriented interface to a procedural operating system, such as the Mach micro-kernel developed by CMU.

Accordingly, Applicants submit that Orton does not describe "a hardware threaded circuit system, comprising: . . . a plurality of processing elements coupled to the task manager, wherein first and second ones of the plurality of processing elements are interconnected for hardware threaded processing to enable dynamic borrowing of processing resources associated with the second one of the plurality of processing elements by the first one of the plurality of processing elements," as recited in claim 18. Therefore, Applicants submit that claim 18 is patentable over Orton and request withdrawal of the art rejections. For at least the same reasons, Applicants submit that claims 19 – 22 are also patentable over Orton and request withdrawal of the art rejections.

Applicant submits that dependent claims 19 - 22 contain additional subject matter that further distinguishes over the art of reference.

With regard to claim 19, Orton fails to anticipate "the circuit maximizes throughput."

With regard to claim 20, Orton fails to anticipate "the circuit reduces power consumption compared to a non-threaded processing for substantially similar system wait times."

With regard to claim 21, Orton fails to anticipate "the first and second processing elements each include a first type of resource and a second type of resource and a multiplexer such that the interconnection includes at least one input signal being provided to the first type of resource in the first and second processing elements."

With regard to claim 22, Orton fails to anticipate "the interconnection includes a connection from an output of the second processing element first type of resource to the first processing element."

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific rejection, issue, or comment does not signify agreement with or concession of that rejection, issue, or comment. In addition, because the arguments made above may not be exhaustive, there may be reasons for withdrawing the prior art cited with regards to any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

Applicants submit that the entire application is now in condition for allowance. Such action is respectfully requested at the Examiner's earliest convenience.

The Examiner is respectfully invited to telephone the undersigning attorney if there are any questions regarding this Response or this application.

The Assistant Commissioner is hereby authorized to charge payment of any additional fees associated with this communication or credit any overpayment to Deposit Account No. 500845, including but not limited to, any charges for extensions of time under 37 C.F.R. §1.136.

Dated: March 25, 2009

Respectfully submitted,

DALY, CROWLEY, MOFFORD & DURKEE, LLP

By: Steven M Cohen

Steven M. Cohen
Reg. No. 59,503
Attorney for Applicant(s)
354A Turnpike Street - Suite 301A
Canton, MA 02021-2714
Tel.: (781) 401-9988, Ext. 126
Fax: (781) 401-9966
smc@dc-m.com

111123_2